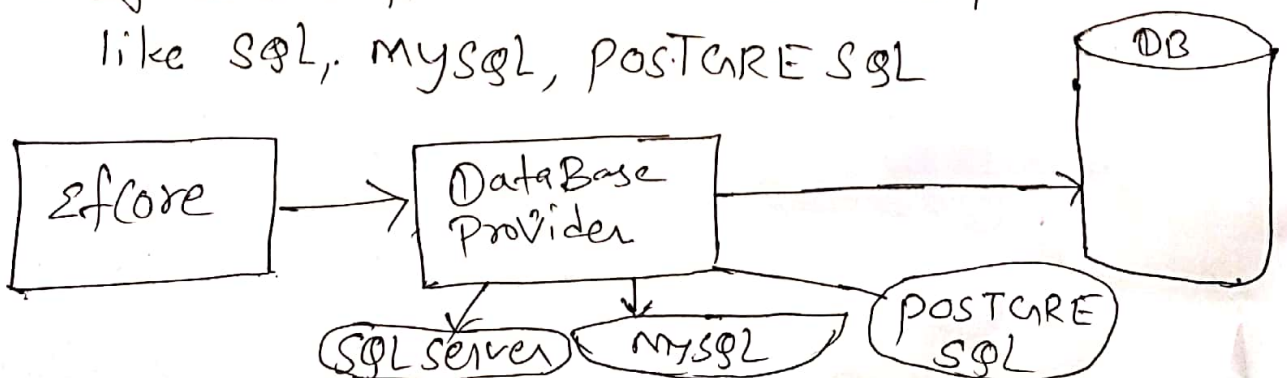


Unit 4 Entity framework and LINQ.

4.1 Introduction to Entity framework

Entity framework Core

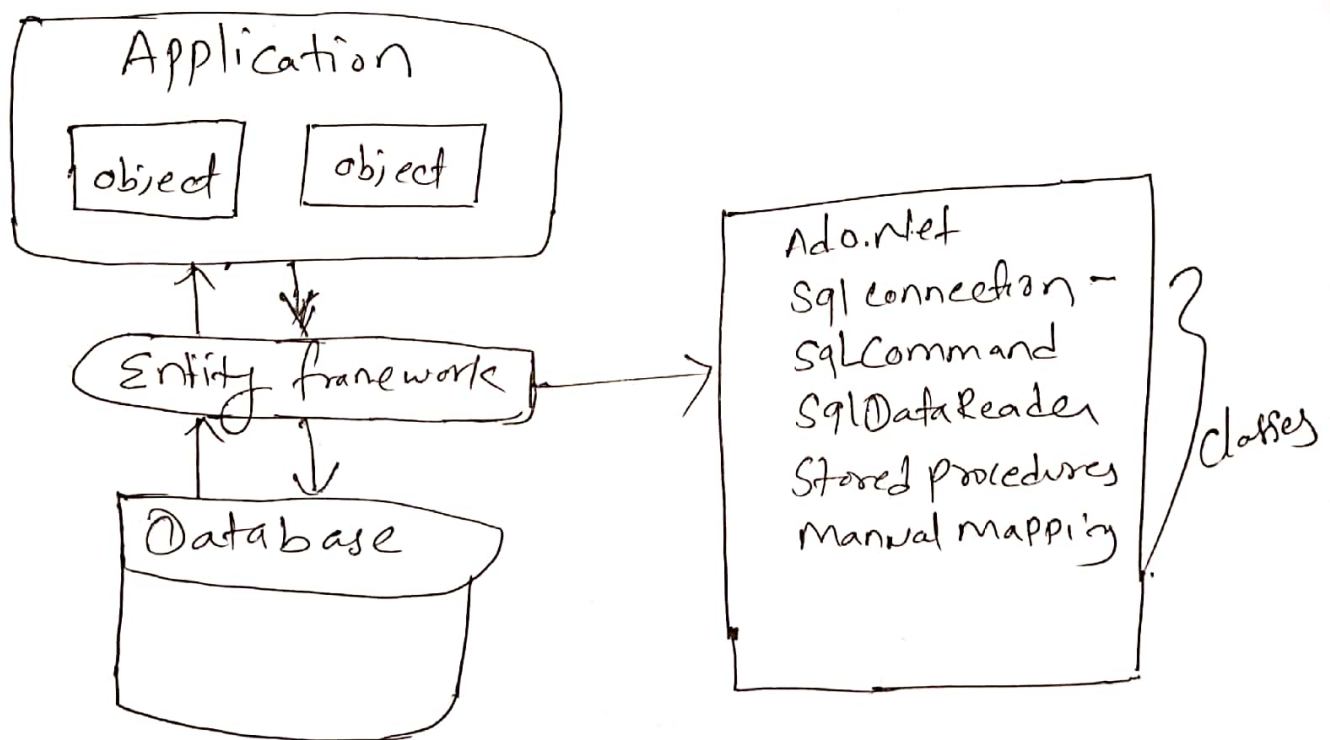
- ↳ Entity framework core is the new version of Entity framework after EF6 but it is redesigned.
- ↳ It is open-source, lightweight, extensible and a cross-platform version of Entity framework.
- ↳ Entity framework core is a data access technology.
- ↳ Entity framework is an object/Relational mapping (ORM) framework.
- ↳ It is an enhancement to ADO.NET that gives developers an automated mechanism for accessing and storing the data in the database.
- ↳ EF Core is intended to be used with .Net Core applications. However, it can also be used with Standard .Net 4.5 + framework based applications.
- ↳ EF Core Development Approaches supports:-
 - 1) Code-first
 - 2) Database-first
- ↳ EF Core supports multiple Database providers like SQL, MySQL, PostgreSQL



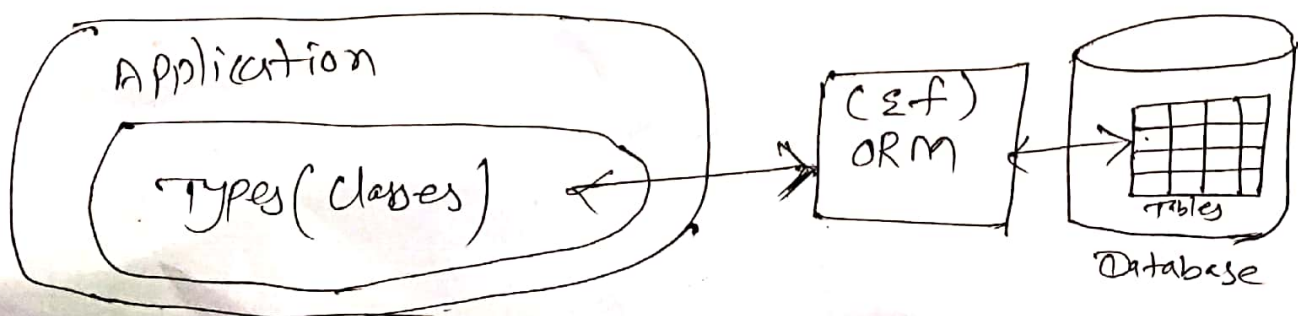
→ ORM is object Relational Mapping framework used to address the data access requirements of ASP.Net Core MVC application.

→ ORM simplifies the process of accessing data from applications.

→ ORM is a tool for storing data from domain objects (model classes) to relational database or vice versa.



Entity framework is the extend version of ADO.NET



- Entity framework (Ef) is an ORM (Object-Relational Mapper) that allows .Net developers to interact with databases using C# objects instead of SQL queries.
- Ef supports CRUD operations (create, Read, update, Delete) efficiently.

Advantages of Entity framework

- Reduces development time
- No need to write SQL queries manually
- Supports different database providers
- Reduces code complexity
- Supports multiple database providers / database approaches (Code first, Database first) etc.

4.2 understanding and Implementing Database Approaches

1. Database first Approach

- ↳ Create the database first, then generate models from it.
- ↳ Entity framework (ORM) Reads the existing database and creates classes (models) automatically.

Steps:

1. Design DB in SQL server
2. use Ef Designer to create model.
3. use generated classes to access data.

2. Code first Approach

↳ create C# classes (models) first, then EF generates the DB from those classes.

Steps:

1. write model classes
2. configure context class (DbContext)
3. EF creates the database.

Advantages: full control over code and design.

3. Model first Approach

↳ Design the model using a designer tool and EF generates both DB and code from that model.

steps:

1. use EF Designer to create model visually.
2. Generate DB and code from the model.

4.3 Linq (Language Integrated Query)

↳ Linq stands for Language Integrated Query.

↳ Linq in C# is used to work with data access from data sources such as objects, data sets, SQL server, and XML etc.



- ↳ LINQ is a data querying API with SQL like query syntaxes.
- ↳ Microsoft's query language is fully integrated and offers easy data access from in-memory objects, collections, databases, XML documents, and many more.
- ↳ LINQ allows writing queries even without the knowledge of query languages like SQL, XML etc.
- ↳ LINQ is a set of extensions, LINQ ably integrates queries in C# and Visual Basic.
- ↳ LINQ provides functions to query data from all kinds of data sources.
- ↳ The data source could be a collection of objects, a database or XML files.
- ↳ LINQ introduced in C# version 3.0 as a major part of .Net framework 3.5 in 2007.

Standard query operators:

- ① from
- ② select
- ③ where
- ④ orderby

Syntax

from the point of view of SQL

select * from table.name
where age > 20

Syntax

from the point of view of LINQ:

from i in age where i > 20 select i

✓ using System;
 using System.Collections.Generic;
 ✓ using System.Linq;
 using System.Text;

namespace LING-Demo

{
 Class program

{
 static void Main (string[] args)

{
 int[] age = { 12, 23, 21, 22, 45, 33, 20, 26,
 55, 66 };



var a = from i in age where i > 20 select
 i;

foreach (int item in a)
 {
 Console.WriteLine(item);
 }
 }
 }
 }

output: 23
 21
 22
 45
 33
 26
 55
 66

Note:

using System.Linq; is
 compulsory when working
 with LINQ statements

If you don't provide this
 namespace, compile time error
 will occur.

Sorting the Result: (Ascending / Descending order)

from the point of view of SQL:

select * from table-name where
age > 20 order by age asc/
desc

from the point of view of LINQ

from i in age where i > 20
orderby i select i

(By Default)
Ascending order

from i in age where i > 20
orderby i descending select i

(for Descending)

Example program

```
using System;  
using System.Linq;  
namespace Linq-Demo
```

```
{  
    class program
```

```
{  
    static void Main (string args [])
```

```
{  
    int[] age = { 12, 23, 21, 25, 45, 33, 20, 26,  
                 55, 66 };
```

```
    var a = from i in age where i > 20 orderby  
            i descending select i;
```

```
    foreach (int item in a)
```

```
{  
        Console.WriteLine (item);
```

```
}  
}  
}
```

output: 66

55

45

33

26

23

22

21

Example 3

```
static void main (string[] args)
```

```
{
```

```
    string[] names = { "Rajesh", "Rakesh", "RAM",  
                        "Harsh", "Nipun" };
```

```
    var a = from name in names where name.  
            Contains("R") select name;
```

```
    foreach (string item in a)
```

```
    {
```

```
        Console.WriteLine(item);
```

```
    }
```

```
}
```

output: Rajesh

Rakesh

RAM

// case sensitive

// uppercase and lowercase are
different

Example 4

```
var a = from name in names where name.  
        StartsWith("H") select name;
```

output: Harsh

If you use this on this line

Advantages of LINQ

- ↳ LINQ offers an object-based, language integrated way to query over data no matter where that data came from. So through LINQ we can query database, XML as well as collections.
- ↳ Compile time syntax checking.
- ↳ LINQ Allows you to query collections like arrays, enumerable classes etc in the native Language of your application, Like VB or C# in much the same way as you would query a database using SQL.
- ↳ User does not need to Learn new query Languages for a different type of data source or data format.
- ↳ It increase the readability of the code.
- ↳ Query can be reused.
- ↳ It provides Intellisense for generic collections.
- ↳ Linq can be used with array or collections.
- ↳ Linq supports filtering, sorting, ording, grouping.
- ↳ LINQ is integrated with C# so it make easy debugging.
- ↳ LINQ is used for working with relational database management system eg SQL server, mysql.
- ↳ LINQ is not just only about querying the data it also allows us to perform CRUD operations Like insert, update, delete and read operations from database.

C - CREATE - INSERT
R - READ - SELECT
U - UPDATE - UPDATE
D - DELETE - DELETE

10

Advantages of using Linq for SQL

Using Linq with SQL Server

1. Linq queries are checked during compile time.

↳ Linq queries are not compiled by database engine, it is compiled by C# because Linq is integrated in C#.

→ Linq queries are executed and verified by Linq query engine which is provided in our .net framework.

↳ Linq queries follow C# syntax.

↳ Linq queries are compiled in native language of a developer like C# or vb.net

2. There is Intellisense support.

↳ Linq queries are executed by C# compiler so the compiler knows the data types of database objects and also displays appropriate intellisense to a developer.

5. 100% object oriented code

3. It is type safe.

↳ In this case, all the data types of columns are already known by the C# compiler that's why you have to pass only appropriate values.

4. Debugging of Linq is possible

↳ Compiler knows the errors and displays on compile time then we can easily debug.

5.1007. object oriented side

<u>Relational Types</u>	<u>object oriented Types</u>
Table	→ Class

11

Column → property

Row → Object/Instance

Stored procedure → Method

If you want to work with LINQ TO SQL, first you have to convert or map all the relational objects of database into object oriented types.

This process is called ORM (Object Relational Mapping)

Using ADO.Net with SQL Server

1. SQL queries are checked during runtime

- Your SQL queries defined by ADO.Net's SqlCommand is executed by SQL server Database engine not by C# compiler.

- When we writing SQL queries using ADO.Net you might have noted that we write queries in double quotes, it means we have write them as a string.

- Strings are not compiled by C#, that string is sent to database engine and executed by SQL server database engine.

- Then Database engine is going to verify that query is right or wrong, If query is wrong it will raise exception because exceptions are raised during runtime.

- As a C# developer, it is not best practice to put the extra burden to database engine.

2. No Intellisense support.

Because all work are done by the database engine so that's why C# compiler did not know what intellisense should be displayed to a developer.

3. It is not type safe.

↳ In this case, you need to know all the columns and its data types.

4. Debugging of SQL Statements will be nightmare.

↳ SQL queries are present in double quotes and are executed on database engine that's why no debugging is possible.

↳ All the code will be sent to the database engine and database engine provide run time errors if errors finds.

5. SQL queries is a combination of object oriented and Relational.